

REMARKS

Claims 1-22 are pending in this application. All independent claims 1, 10 and 19 have been
5 amended.

Rejections of claims 1-22 under 35 USC Sec. 102 and 103 are now moot because the Applicant
amended the claims in order to define the invention more particularly. All independent claims 1,
10 and 19 are now specifically directed to show an improvement of a standard database
management system which includes a direct call mechanism replacing the lookup function of a
run-time interpreter and a method for pre-processing an access plan to provide a direct call
mechanism in such a system. They recite novel structure and thus distinguish over the cited prior
art, under 35 U.S.C. 102(e) and 103.

3. Claims 1-3, 6-7, 10-12, 15-16 and 19 stand rejected under 35 U.S.C. 102(e) as being
15 anticipated by US Patent No. 5, 386,557 issued to Boykin et al. (herein Boykin).

The Examiner stated that, with respect to claim 1, Boykin discloses (a) determining from
the access plan an executable function associated with a first operation code (access plan
contains low-level interpreted code which has been compiled from database access language
such as SQL and the thread, item 50, containing code for performing the function of the SQL
20 statement (see fig. 1, and fig. 2, col. 3, lines 60-67 and col. 4, lines 20-25; also see col. 5, lines
35-49); and (b) augmenting said first operation code in the access plan with a pointer to said
executable function (item 54 IUDO containing pointers to any data values being necessary to
perform the execution of the operations: col. 4, lines 44-65 and col. 6, lines 36-42).

25 The Examiner stated that, with respect to claim 2, Boykin discloses the remaining
operation codes in the access plan (col. 5, lines 35-62).

The Examiner stated that, with respect to claim 3, Boykin discloses augmenting said first

operation code in the access plan with a pointer to an intermediate function, said intermediate function including a data structure for storing a pointer to said executable function (col. 4, lines 20-32 and lines 44-65).

5 The Examiner stated that, with respect to claim 6, Boykin discloses assessing the executable function associated with the first operation code and if applicable, replacing the call to the executable function with a call to a second executable function (col. 9, lines 2-7).

The Examiner stated that, with respect to claim 7, Boykin discloses intermediate function which includes processing operations for the first operation code or the executable function associated with the first operation code (col. 4, lines 44-65).

10 The Examiner stated that claim 10 is essentially the same as claim 1 except that it is directed to a computer program product rather than a method (see fig. 1, and fig. 2, col. 3, lines 60-67 and col. 4, lines 20-25; also see col. 5, lines 35-49; and item 54 IUDO containing pointers to any data values being necessary to perform the execution of the operations: col. 4, lines 44-65 and col. 6, lines 36-42), and is rejected for the same reason as applied to the claim 1 hereinabove.

15 The Examiner stated that claim 11 is essentially the same as claim 2 except that it is directed to a computer program product rather than a method (col. 5, lines 35-62), and is rejected for the same reason as applied to the claim 2 hereinabove.

20 The Examiner stated that claim 12 is essentially the same as claim 3 except that it is directed to a computer program product rather than a method (col. 4, lines 20-32), and is rejected for the same reason as applied to the claim 3 hereinabove.

The Examiner stated that claim 15 is essentially the same as claim 6 except that it is directed to a computer program product rather than a method (col. 9, lines 2-7), and is rejected for the same reason as applied to the claim 6 hereinabove.

The Examiner stated that claim 16 is essentially the same as claim 7 except that it is directed to a computer program product rather than a method (col. 4, lines 44-65), and is rejected for the same reason as applied to the claim 7 hereinabove.

5 The Examiner stated that claim 19 is essentially the same as claim 1 except that it is directed to a system rather than a method (see fig. 1, and fig. 2, col. 3, lines 60-67 and col. 4, lines 20-25; also see col. 5, lines 35-49; and item 54 IUDO containing pointers to any data values being necessary to perform the execution of the operations: col. 4, lines 44-65 and col. 6, lines 36-42), and is rejected for the same reason as applied to the claim 1 hereinabove.

10 The claims of the present invention recite novel structure and thus distinguish over the cited prior art, under 35 U.S.C. 102(e). They specifically describe an improved database management system and method capable of providing a direct call mechanism, thus replacing the existing lookup function of a run-time interpreter, as described in Specification p. 2, li. 16-31 and p. 6, li. 14-30, to provide faster access which is cost-effective. These features are recited in claims 1-22
15 and throughout the Abstract and Specification, including Summary, and are not shown in the Boykin reference which does not show all the features of the present invention, operating in the same way and for the same purpose.

Reference Boykin teaches a system for enforcing referential constraints when changes are made
20 to a database, where the RDBMS does not interpret database access commands at run time (col. 1, li. 19-23). Moreover, according to col. 1, li. 55-68, the code in Boykin reference is inserted directly into the access plan at compile time. Therefore, Boykin reference does not have the steps (a) - (b) of the independent claims of the present invention. It is from a completely different field, has different components, works in a different mode of use and produces different results.
25 Boykin reference does not show, teach or suggest an improvement of a standard database management system and a method for replacing a lookup function of a run-time interpreter with a direct call mechanism that integrates easily into an existing RDBM system and is cost-effective, which was a long-felt need in this field.

The Examiner stated that, as per claim 1-3, 6-7, 10-12, 15-16 and 19, the Boykin reference in cols. 3-6 and 9 shows augmenting of an access plan to IUDO pointers. However, these columns and col 10 li. 9-17 in Boykin describe a feature wherein the IUDOs for inserts, deletes and updates are made at the time the access plan is compiled and the appropriate subthreads are inserted by the compiler, as shown in col. 5, li. 5-12 and 35-56. Such complexity is not needed in the present invention and this technique cannot be applied to it. Moreover, these lines teach away from the present invention which does not compile the code but pre-processes the access plan to provide a direct call at run-time, the feature that does not exist in Boykin reference.

Therefore, all independent claims 1, 10 and 19 and all claims dependent upon these in the present invention recite novel structure and thus distinguish over the cited prior art, Boykin, and are not anticipated by it under 35 U.S.C. 102(e).

Claims 4-5, 8-9, 13-14, 17-18 and 20-22 stand rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,386,557 issued to Boykin et al. (herein Boykin) in view of US Patent No. 5,822,750 issued to Jou et al. (herein Jou).

The Examiner stated that, with respect to claims 4-5, Boykin discloses the method as discussed in claim 1. It is noted with appreciation that the Examiner held that Boykin does not explicitly indicate, "data structure includes means for storing information and data structure providing means for storing information" but that Jou discloses data structure for storing information as claimed (col. 10, lines 10-67, col. 11, lines 1-14 and also see col. 3, lines 20-36) and that, therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Boykin with the teachings of Jou so as to obtain a method for pre-processing access plan from a query, and that this combination would provide a method enabling to optimize derived table evaluation and to eliminate the table access operations in evaluation of a query increases efficiency (Jou - col. 8, lines 60-65) in the relational database management environment.

The Examiner stated that, with respect to claims 8-9, Boykin discloses the method as discussed in claim 1. It is noted with appreciation that the Examiner held that Boykin does not

explicitly indicate, "gathering statistics on the use of the executable function; and a pause for receiving user input before or after the call to the executable function" and that Jou discloses gathering statistics data (col. 11, lines 38-51; also col. 12, lines 32-67 and col. 13, lines 1-15) and receiving user input as claimed (col. 12, lines 18-45) and that, therefore, it would have been
5 obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Boykin with the teachings of Jou so as to obtain a method for pre-processing access plan from a query and that this combination would provide a method enabling to optimize derived table evaluation and to eliminate the table access operations in evaluation of a query increases efficiency (Jou - col. 8, lines 60-65) in the relational database management
10 environment.

The Examiner stated that claims 13-14 are essentially the same as claims 4-5 except that it is directed to a computer program product rather than a method (col. 10, lines 10-67, col. 11, lines 1-14 and also see col. 3, lines 20-36), and are rejected for the same reason as applied to the claims 4-5 hereinabove.

15 The Examiner stated that claims 17-19 are essentially the same as claims 8-9 except that it is directed to a computer program product rather than a method (col. 11, lines 38-51; also col. 12, lines 32-67 and col. 13, lines 1-15; and col. 12, lines 18-45), and are rejected for the same reason as applied to the claims 8-9 hereinabove.

The Examiner stated that, with respect to claims 20-22, Boykin discloses the method as
20 discussed in claim 19 and that it discloses adding another pointer (see fig 2, col. 4, lines 20-34 and 6, 36-42). It is noted with appreciation that the Examiner held that Boykin does not explicitly indicate, "replacing said operation codes; storing information and data structure" and that Jou discloses replacing operation as claimed col. 14, lines 26-45); data structure for storing information as claimed (col. 10, lines 10-67, col. 11, lines 1-14 and also see col. 3, lines 20-36)
25 and that, therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Boykin with the teachings of Jou so as to

obtain a method for pre-processing access plan from a query and that this combination would provide a method enabling to optimize derived table evaluation and to eliminate the table access operations in evaluation of a query increases efficiency (Jou - col. 8, lines 60-65) in the relational database management environment.

5 Claims 4-5, 8-9, 13-14, 17-18 and 20-22 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Boykin in view of Jou. However, since it has been shown above that all independent claims 1, 10 and 19 and all claims dependent upon these in the present invention recite novel structure and thus distinguish over the cited prior art, Boykin, and are not anticipated
10 by it under 35 U.S.C. 102(e), the combination of Boykin and Jou does not make the present invention obvious and unpatentable under Section 103 for the same reasons stated above.

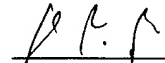
Jou reference is directed to optimization of correlated queries and has nothing in common with the present invention, according to the Examiner, except the data structure for storing
15 information and gathering statistic data. Thus, neither Jou nor Boykin teach improving a RDBMS by replacing a lookup function of a run-time interpreter with a direct call mechanism. Therefore, the present invention is created to solve problems from a different art field than Boykin and Jou. Moreover, Boykin and Jou references do not perform any elements of the independent claims 1, 10, and 19, and their dependent claims. Therefore, the combination of
20 Boykin and Jou fails to teach all elements of claims 1, 10 and 19 and their dependent claims. Moreover, Boykin and Jou satisfy a different need from a different area and do not teach optimization during run time. Therefore, Boykin and Jou references cannot be used to invalidate independent claims 1, 10, and 19 and their dependent claims.

25 Therefore, all submitted claims are allowable over the cited reference and their reconsideration is respectfully requested. None of the cited references discloses the subject matter and features of claims 1-22 of the present invention and, even if they did show some individual features, they would not be able to meet the claims of the present invention which provide new and unexpected results over these references and are thus not anticipated under Section 102 and unobvious and
30 patentable under Section 103.

In view of the above, it is submitted that this application is now in good order for allowance, which applicant respectfully solicits. Should matters remain which the Examiner believes could be resolved in a telephone interview, the Examiner is kindly requested to telephone the applicant's undersigned attorney. No additional fee is required in connection with this communication since the Amendment is mailed within three months from the Office Action and the number of claims is not extending the original number of claims. However, any underpayment is authorized to be charged to Deposit Account Number 09-0460 in the name of IBM Corporation.

Respectfully submitted,

Date: January 31, 2003



Sandra M. Parker
Reg. No. 36,233

LAW OFFICE OF SANDRA M. PARKER
329 La Jolla Avenue
Long Beach, CA 90803
Phone: (562) 597-7504
FAX: (562) 597-1841.

AMENDED CLAIMS SHOWING INSERTIONS AND DELETIONS

1. (Amended) A method for pre-processing an access plan generated for a query in a relational database management system to include a direct call mechanism replacing a lookup function of a run-time interpreter, said access plan including a plurality of operation codes, each of said operation codes being associated with one or more executable functions for performing the query, said method comprising the steps of:

(a) determining from the access plan an executable function associated with a first operation code; and

(b) augmenting said first operation code in the access plan with a pointer to said executable function to provide a direct call mechanism.

10. (Amended) A computer program product for use on a computer wherein queries are entered by a user for retrieving data in a relational database management system having a query optimizer for generating an access plan for executing the query, said query optimizer including a direct call mechanism replacing the lookup function of a run-time interpreter, said computer program product comprising:

a recording medium;

means recorded on said recording medium for instructing said computer to perform the steps of:

(a) determining an executable function associated with a first operation code in the access plan, the first operation code being one of a plurality of operation codes; and

(b) augmenting said first operation code in the access plan with a pointer to said

12 executable function to provide a direct call mechanism.

1 19. **(Amended)** A relational database management system for use with a computer system
2 wherein queries are entered by a user for retrieving data from tables, the relational database
3 management system including a query optimizer for generating an access plan associated with
4 the queries entered by the user, said query optimizer including a direct call mechanism replacing
5 a lookup function of a run-time interpreter, said relational database management system
6 comprising:

7 (a) means for determining an executable function associated with each of a plurality
8 of operation codes in the access plan; and

9 (b) means for augmenting said operation codes in the access plan with a pointer to
10 said executable function associated with each operation code to provide a direct call mechanism.